

WHAT IS CLAIMED IS:

1. A method of parsing, comprising:

receiving an XML stream;

parsing said XML stream

on encountering a parent element in said XML stream, utilising an XML schema to locate a type for said parent element;

where said type is a complex type, determining whether a mixed flag for said complex type in said schema is set to true;

where said mixed flag is set to true, interpreting fragments embedded in said parent element in accordance with said complex type, each fragment being one of an arbitrary string and an element;

where, in accordance with said complex type, an embedded fragment corresponds to a dummy element having a simple type, with one of (i) a name of said dummy element and (ii) a name of said simple type being one of a predetermined set of names, interpreting said embedded fragment as one of an embedded simple type and an inherited simple type.

2. The method of claim 1 wherein said embedded fragment is considered to correspond to a dummy element when minimum and maximum occurrences of an element in said schema corresponding with said embedded fragment are set to zero.

3. The method of claim 1 wherein said embedded fragment is considered to correspond to a dummy element when an element in said schema corresponding with said embedded fragment is incorporated in a local group, said local group having its minimum and maximum occurrences set to zero.

4. The method of claim 3 wherein said embedded fragment is interpreted as an embedded simple type and further comprising using minimum and maximum occurrences of said dummy element to determine whether said dummy element is mandatory or optional.

5. The method of claim 2 wherein said embedded fragment is interpreted as an embedded simple type and further comprising using an application annotation associated with said dummy element to determine whether said dummy element is mandatory or optional.

5 6. The method of claim 2 wherein said complex type has a sequence construct, said dummy element appearing in a pre-selected position within said sequence construct, whereby the position of an instance of said dummy element in an XML stream may be specified in said schema.

10 7. The method of claim 2 wherein said simple type name is one of said predetermined set of names, said simple type extending a basic simple type.

8. The method of claim 7 wherein said basic simple type is one of a string, integer, floating point number, date, time, decimal number.

15 9. The method of claim 2 wherein said XML schema specifies the position of an instance of said dummy element in any XML stream of said complex type, and further comprising using said XML schema to locate said embedded element that corresponds to said dummy element.

20 10. The method of claim 9 wherein said simple type is an inherited simple type and said dummy element is located as the first element in said complex type.

11. A parser, comprising:

means for receiving an XML stream;

25 means for parsing said XML stream

means for, on encountering a parent element in said XML stream, utilising an XML schema to locate a type for said parent element;

means for, where said type is a complex type, determining whether a mixed flag for said complex type in said schema is set to true;

means for, where said mixed flag is set to true, interpreting fragments embedded in said parent element in accordance with said complex type, each fragment being one of an arbitrary string and an element;

means for, where, in accordance with said complex type, an embedded fragment corresponds to a dummy element having a simple type, with one of (i) a name of said dummy element and (ii) a name of said simple type being one of a predetermined set of names, interpreting said embedded fragment as an embedded simple type.

12. The parser of claim 11 wherein said means for interpreting said embedded fragment as an embedded simple type.determines an embedded fragment corresponds to a dummy element when minimum and maximum occurrences of an element in said schema corresponding with said embedded fragment are set to zero.

13.. The parser of claim 11 wherein said means for interpreting said embedded fragment as an embedded simple type.determines said embedded fragment corresponds to a dummy element when an element in said schema corresponding with said embedded fragment is incorporated in a local group, said local group having its minimum and maximum occurrences set to zero.

14. The parser of claim 13 further comprising means for, where said embedded fragment is interpreted as an embedded simple type, using minimum and maximum occurrences of said dummy element to determine whether said dummy element is mandatory or optional.

15. The parser of claim 12 further comprising means for, where said embedded fragment is interpreted as an embedded simple type, using an application annotation associated with said dummy element to determine whether said dummy element is mandatory or optional.

16. The parser of claim 12 wherein said simple type name is one of said predetermined set of names, said simple type extending a basic simple type.

17. The parser of claim 16 wherein said basic simple type is one of a string, integer, floating point number, date, time, and decimal number.

18. The parser of claim 12 wherein said XML schema specifies the position of an instance of said dummy element in any XML stream of said complex type, and further comprising means for using said XML schema to locate said embedded element that corresponds to said dummy element.

19. The parser of claim 18 wherein said simple type is an inherited simple type and said dummy element is located as the first element in said complex type.

20. An XML schema, comprising:

at least one complex type, said complex type having a mixed flag set to true and including an element set to be a dummy element, said dummy element having a simple type, one of (i) a name of said dummy element and (ii) a name of said simple type being one of a predetermined set of names for use with a parser which, on parsing said schema, will interpret said dummy element as one of an embedded simple type and an inherited simple type.

21. The schema of claim 20 wherein said dummy element is set to be a dummy element by reason of having its minimum and maximum occurrences set to zero.

22. The schema of claim 20 wherein said dummy element is set to be a dummy element by reason of incorporation in a local group, said local group having its minimum and maximum occurrences set to zero.

23. The schema of claim 22 wherein when said dummy element is interpreted as an embedded simple type, said dummy element having its minimum and maximum occurrences set to indicate whether said dummy element is mandatory or optional.

24. The schema of claim 21 wherein when said dummy element is interpreted as an embedded simple type, said dummy element having an associated application annotation, said application annotation set to indicate whether said dummy element is mandatory or optional.

25. The schema of claim 20 wherein said complex type has a sequence construct, said dummy element appearing in a pre-selected position within said sequence construct, whereby the position of an instance of said dummy element in an XML stream may be specified in said schema.

26. The schema of claim 20 wherein said simple type name is one of said predetermined set of names, said simple type extending a basic simple type.

27. The schema of claim 26 wherein said basic simple type is one of a string, integer, floating point number, date, time, decimal number.

28. A computer readable medium containing computer executable instructions which, when executed by a processor, cause said processor to undertake the method of any of claims 1 to 10.